

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Hyun-Kwon CHUNG et al

Application No.: Unassigned

Group Art Unit: Unassigned

Filed: December 5, 2003

Examiner: Unassigned

For: APPARATUS AND METHOD FOR EXECUTING APPLLET

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Korean Patent Application No(s). 2002-77280

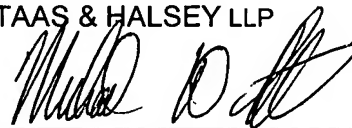
Filed: December 6, 2002

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

By:



Michael D. Stein
Registration No. 37,240

Date: December 5, 2003

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 10-2002-0077280
Application Number

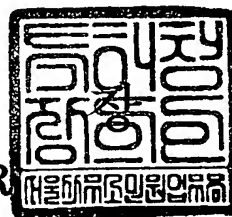
출원 년 월 일 : 2002년 12월 06일
Date of Application DEC 06, 2002

출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 11 월 11 일

특 허 청
COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0010
【제출일자】	2002. 12. 06
【국제특허분류】	G06F
【발명의 명칭】	애플릿 실행 장치 및 방법
【발명의 영문명칭】	Apparatus and method for executing an applet
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	1999-009556-9
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2000-002816-9
【발명자】	
【성명의 국문표기】	정현권
【성명의 영문표기】	CHUNG, Hyun Kwon
【주민등록번호】	721217-1042731
【우편번호】	464-800
【주소】	경기도 광주군 광주읍 탄벌리 동보아파트 104동 906호
【국적】	KR
【발명자】	
【성명의 국문표기】	허정권
【성명의 영문표기】	HEO, Jung Kwon
【주민등록번호】	681207-1830616

【우편번호】 137-766
 【주소】 서울특별시 서초구 반포2동 주공아파트 2단지 203동 504호
 【국적】 KR
 【발명자】
 【성명의 국문표기】 박성욱
 【성명의 영문표기】 PARK,Sung Wook
 【주민등록번호】 710327-1041719
 【우편번호】 137-073
 【주소】 서울특별시 서초구 서초3동 1595-2 센쥬리오피스텔 2동 1207호
 【국적】 KR
 【발명자】
 【성명의 국문표기】 정길수
 【성명의 영문표기】 JUNG,Kil Soo
 【주민등록번호】 750903-1917317
 【우편번호】 445-970
 【주소】 경기도 화성군 태안읍 병정 한신아파트 107동 707호
 【국적】 KR
 【취지】 특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인
 이영필 (인) 대리인
 이해영 (인)
 【수수료】
 【기본출원료】 20 면 29,000 원
 【가산출원료】 18 면 18,000 원
 【우선권주장료】 0 건 0 원
 【심사청구료】 0 항 0 원
 【합계】 47,000 원
 【첨부서류】 1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 인터랙티브 콘텐츠를 재생하는 재생장치에 있어서 마크업 문서에 연결된 애플릿을 실행시키는 장치 및 방법에 관한 것이다. 그 애플릿 실행 방법은 (a)애플리케이션 매니저가 애플릿 실행을 요청받는 단계; (b)상기 애플리케이션 매니저가 상기 애플릿이 바운드 애플릿인지 언바운드 애플릿인지 애플릿 종류를 판별하는 단계; (c)상기 애플리케이션 매니저가 메모리에 저장된 상기 애플릿을 버추얼 머신에 로딩하는 단계; 및 (d) 상기 (b)단계에서의 판별 결과, 상기 애플릿이 언바운드 애플릿이면, 상기 애플리케이션 매니저가 상기 버추얼 머신에게 통지하여 상기 애플릿이 초기(Init) 상태가 되도록 제어하는 단계; (e)상기 애플릿이 초기(Init) 상태가 된 직후, 상기 애플리케이션 매니저가 상기 버추얼 머신에게 통지하여 상기 애플릿이 스타트(Start) 상태가 되도록 제어하는 단계를 포함한다.

본 발명에 따른 애플릿 실행 장치 및 방법은, 애플릿의 종류를 나누고 각각의 라이프 사이클을 달리 정의하여 애플릿의 기능을 확장시키고, 특히 마크업 문서의 전환 여부와 무관하게 계속적으로 동작하는 애플릿을 제공한다.

【대표도】

도 12

【명세서】

【발명의 명칭】

애플릿 실행 장치 및 방법{Apparatus and method for executing an applet}

【도면의 간단한 설명】

도 1은 인터랙티브 콘텐츠를 재생하기 위한 시스템을 설명하기 위한 블록도,

도 2는 브라우저를 통해 AV 화면과 마크업 문서 화면이 함께 디스플레이되는 것을 나타내는 도면,

도 3은 콘텐츠 저장매체에 저장된 데이터의 디렉토리 구조를 보여주기 위한 도면,

도 4는 브라우저를 통해 화면에 출력되는 마크업 문서의 구성요소들을 나타내는 도면,

도 5는 도 1에 도시된 인터랙티브 콘텐츠 재생장치의 상세 구성을 설명하기 위한 블록도,

도 6은 도 5에 도시된 프리젠테이션 엔진의 상세 구성을 설명하기 위한 블록도,

도 7은 도 6에 도시된 JVM의 상세 구성을 설명하기 위한 블록도,

도 8은 도 6에 도시된 브라우저의 상세 구성을 설명하기 위한 블록도,

도 9는 JVM, 브라우저 및 애플리케이션 매니저간의 상호 동작을 나타내는 도면,

도 10은 바운드 애플릿의 라이프 사이클을 나타내는 도면,

도 11은 언바운드 애플릿의 라이프 사이클을 나타내는 도면,

도 12는 본 발명에 따른 애플릿 실행 방법을 설명하기 위한 흐름도,

도 13a 내지 도 13e는 본 발명에 따른 바운드 애플릿과 언바운드 애플릿의 예를 나타내는 도면이다.

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <14> 본 발명은 애플릿 실행 장치 및 방법에 관한 것으로서, 특히 인터랙티브 콘텐츠를 재생하는 재생장치에 있어서 인터랙티브 기능을 지원하기 위한 마크업 문서에 연결된 애플릿을 실행시키는 장치 및 방법에 관한 것이다.
- <15> 인터랙티브 콘텐츠란 정보기록매체 예컨대, 인터랙티브 DVD에 Audio/Video 데이터(이하, AV 데이터라 함) 및 인터랙티브 기능을 지원하기 위한 마크업 문서가 함께 저장된 데이터를 말한다. 마크업 문서는 HTML(Hyper Text Markup Language)이나 XML(eXtended Markup Language) 등의 마크업 언어로 작성된 문서를 말한다.
- <16> 인터랙티브-DVD에 기록된 AV 데이터는 일반 DVD-Video와 동일한 방식으로 재생될 수도 있으며, 브라우저(browser)를 통해 AV 데이터가 재생되어 표시되는 AV 화면이 마크업 문서에 정의된 표시창에 매립되어 마크업 문서와 함께 디스플레이될 수도 있다. 후자가 인터랙티브 기능이 지원되는 경우이다. 인터랙티브 기능은 예를 들면, AV 데이터가 영화 타이틀일 경우 표시창의 한편에서는 영화가 상영되고 다른 한편에는 영화의 자막이 디스플레이되고 또 다른 한편에는 예고편을 광고하는 정지영상 등 다양한 인터랙티브 콘텐츠가 표시되는 등이 있다.
- <17> 한편, 마크업 문서에는 애플릿(applet)이 연결되어 있으며, 애플릿에 의해 다양한 인터랙티브 기능이 구현된다.
- <18> 애플릿이란 작은 응용프로그램을 의미한다. 월드와이드웹(World Wide Web) 상에서는 자바(Java)와 같은 객체지향 프로그래밍 언어를 써서 웹 페이지 즉, 마크업 문서와 함께 사용자

측으로 보내질 수 있도록 작게 만든 프로그램을 애플릿이라고 부른다. 자바 애플릿은 애니메이션이나, 간단한 계산 그리고 사용자가 서버에 별도의 요청을 하지 않고서도 수행할 수 있는 간단한 작업들을 수행할 수 있다.

<19> 종래에는 모든 애플릿들이 연결된 마크업 문서에 종속적이어서 마크업 문서가 언로드(unload)되고 다른 문서로 전환되면, 마크업 문서에 연결된 애플릿들도 동작을 중단해야 했으며, 사용자에 의해 마크업 문서가 다시 호출되어 화면으로 출력되면 연결된 애플릿들도 다시 동작을 하였다.

【발명이 이루고자 하는 기술적 과제】

<20> 따라서 본 발명이 이루고자 하는 기술적 과제는, 마크업 문서에 종속적인 애플릿과 함께 마크업 문서에 종속적이지 아니한 애플릿을 실행하는 장치 및 방법을 제공하는데 있다.

【발명의 구성 및 작용】

<21> 상기 과제를 이루기 위해, 본 발명에 의한 애플릿 실행 장치는,

<22> 외부로부터 입력된 마크업 문서를 저장하는 메모리; 애플릿을 실행하는 버추얼 머신; 상기 메모리로부터 상기 마크업 문서를 입력받아, 상기 마크업 문서에 포함된 상기 마크업 문서와 연관된 애플릿에 관한 정보를 출력하는 브라우저; 및

<23> 상기 브라우저로부터 상기 애플릿에 관한 정보를 입력받아 상기 애플릿을 외부로부터 가져와 상기 메모리에 저장하도록 제어하고, 상기 브라우저로부터 상기 메모리에 저장된 상기 애플릿의 실행을 요청받아, 상기 메모리에 저장된 상기 애플릿을 상기 버추얼 머신에 로딩하고, 상기 애플릿이 바운드 애플릿인지 언바운드 애플릿인지 상기 애플릿의 종류를 판별하여, 상기 애플릿이 언바운드 애플릿이면, 상기 버추얼 머신에게 통지하여 상기 애플릿이 초기(Init) 상

태가 되도록 제어하고, 상기 애플릿이 초기(Init) 상태가 된 직후, 상기 버추얼 머신에게 통지하여 상기 애플릿이 스타트(Start) 상태가 되도록 제어하는 애플리케이션 매니저를 포함한다.

<24> 상기 과제를 이루기 위해, 본 발명에 의한 애플릿 실행 방법은,

<25> (a) 애플리케이션 매니저가 애플릿 실행을 요청받는 단계; (b) 상기 애플리케이션 매니저가 상기 애플릿이 바운드 애플릿인지 언바운드 애플릿인지 애플릿 종류를 판별하는 단계; (c) 상기 애플리케이션 매니저가 메모리에 저장된 상기 애플릿을 버추얼 머신에 로딩하는 단계; 및 (d) 상기 (b)단계에서의 판별 결과, 상기 애플릿이 언바운드 애플릿이면, 상기 애플리케이션 매니저가 상기 버추얼 머신에게 통지하여 상기 애플릿이 초기(Init) 상태가 되도록 제어하는 단계; (e) 상기 애플릿이 초기(Init) 상태가 된 직후, 상기 애플리케이션 매니저가 상기 버추얼 머신에게 통지하여 상기 애플릿이 스타트(Start) 상태가 되도록 제어하는 단계를 포함한다.

<26> 이하, 첨부된 도면들을 참조하여 본 발명에 따른 바람직한 실시예를 상세히 설명한다.

<27> 도 1은 인터랙티브 콘텐츠를 재생하기 위한 시스템을 설명하기 위한 블록도이다. 도 1을 참조하면 인터랙티브 콘텐츠를 재생하기 위한 시스템은 콘텐츠 저장매체(100), 인터랙티브 콘텐츠 재생장치(200), 디스플레이부(300), 리모콘(400) 및 인터넷(500)을 포함한다.

<28> 콘텐츠 저장매체(100)는 인터랙티브 콘텐츠를 저장하는 매체로서 예컨대 인터랙티브 DVD가 있다. 인터랙티브 DVD는 AV 데이터, 마크업 문서 데이터 및 기타 데이터를 저장한다.

<29> 도 3은 콘텐츠 저장매체(100)에 저장된 데이터의 디렉토리 구조를 보여주기 위한 도면으로서, AV 데이터를 저장하는 VIDEO_TS 디렉토리, 마크업 문서 데이터 등의 인터랙티브 기능을 지원하는 데이터가 저장된 DVD_ENAV 디렉토리 및 기타 파일의 구조를 가지고 있다.

- <30> 인터랙티브 콘텐츠 재생장치(200)는 콘텐츠 저장매체(100)에 기록된 AV 데이터를 일반적인 방식으로 재생할 수도 있으며, 도 2에 도시된 바와 같이 브라우저를 통해 AV 데이터가 재생되어 표시되는 AV 화면이 마크업 문서에 정의된 마크업 문서 화면에 매립되어 마크업 문서와 함께 디스플레이할 수도 있다.
- <31> 또한 인터랙티브 콘텐츠 재생장치(200)는 인터넷(500)과 같은 네트워크를 통해 인터랙티브 콘텐츠를 제공받아 재생할 수도 있다. 예를 들면, 인터랙티브 콘텐츠 재생장치(200)를 이용하여 인터넷(500)을 통해 온라인 게임을 즐기는 경우이다.
- <32> 도 4는 브라우저를 통해 화면에 출력되는 마크업 문서의 구성요소들을 나타내는 도면이다. 마크업 문서를 구성하는 요소로는 마크업 문서와 연결된 다수의 애플릿들, GIF, JPEG과 같은 정지영상 및 프레임이 있다.
- <33> 도 5는 도 1에 도시된 인터랙티브 콘텐츠 재생장치(200)의 상세 구성을 설명하기 위한 블록도이다.
- <34> 도 5를 참조하면 인터랙티브 콘텐츠 재생장치(200)는 리더(reader)(210), 버퍼 메모리(220), 캐쉬 메모리(230), 디코더(240), 프리젠테이션 엔진(250), 네트워크 데이터 트랜시버(260) 및 블렌더(270)를 포함한다.
- <35> 리더(210)는 픽업부(도시되지 않음)를 구비하여 콘텐츠 저장매체(100)로부터 데이터를 독출하여 버퍼 메모리(220)로 AV데이터를 출력하고, 캐쉬메모리(230)로는 인터랙티브 기능에 관련된 마크업 문서 데이터 및 연관된 애플릿 등의 데이터를 출력한다.
- <36> 디코더(240)는 AV데이터를 입력받아 복호화한다. 예컨대, MPEG방식으로 부호화된 비디오 데이터를 복호화하거나 MPEG, AC3 등의 방식으로 부호화된 오디오 데이터를 복호화한다.

- <37> 프리젠테이션 엔진(250)은 마크업 문서 등의 데이터를 입력받아 마크업 문서를 해석하여 블렌더(270)로 출력한다. 또한 리모콘(400)을 통해 사용자로 부터 입력 UOP(User OPeration)을 받아 인터랙티브한 기능을 수행한다. UOP는 예컨대 사용자가 리모콘을 통해 재생, 재생 정지, 재생 일시 정지 등의 재생에 관한 명령과 그 외 인터랙티브 콘텐츠의 재생중에 필요한 네비게이션 키 입력 등에 관한 명령을 포함한다.
- <38> 네트워크 데이터 트랜시버(260)는 인터넷(500)을 통해 원격지 서버(도시되지 않음)와의 통신을 수행한다.
- <39> 도 6은 도 5에 도시된 프리젠테이션 엔진(250)의 상세 구성을 설명하기 위한 블록도이다. 도 6을 참조하면, 프리젠테이션 엔진(250)은 JVM(251), 브라우저(253) 및 애플리케이션 매니저(255)를 포함한다.
- <40> JVM(251)은 자바 버추얼 머신(Java Virtual Machine)을 말한다. JVM(251)은 마크업 문서에 연결된 자바 애플릿을 실행한다. 물론, 다른 프로그램 방식에 의한 애플릿을 실행하는 버추얼 머신도 가능하다. 본 실시예에서는 자바 애플릿과 자바 버추얼 머신을 예로 하여 설명한다.
- <41> 브라우저(253)는 마크업 문서를 디스플레이 장치(300)를 통해 디스플레이한다.
- <42> 애플리케이션 매니저(255)는 JVM(251) 및 브라우저(253)와 같은 애플리케이션의 동작을 관리한다. 애플리케이션 매니저(255)는 독립적인 메니징 프로그램일 수도 있으며 오퍼레이팅 시스템(Operating System:OS)의 일부로 구현될 수도 있다.
- <43> 도 7은 도 6에 도시된 JVM(251)의 상세 구성을 설명하기 위한 블록도이다. 도 7을 참조하면, JVM(251)은 컴파일러(251a), 인터프리터(251b) 및 런타임 시스템(251c)을 포함한다.

- <44> 자바 프로그래밍 언어로 작성된 프로그램 즉, 자바 애플릿은 바이트 코드(byte code) 형태로 컴파일러(251a) 또는 인터프리터(251b)에 입력되고 컴파일러(251a) 또는 인터프리터(251b)에서 바이트 코드가 컴파일링 또는 인터프리팅되어 기계어 명령들이 런타임 시스템(251c)으로 출력되어, 런타임 시스템(251c)은 자바 애플릿을 실행시킨다.
- <45> 버추얼 머쉬, 특히 자바 버추얼 머신(JVM)(251)에 대한 상세한 설명은 "The Java Virtual Machine Specification"(Time Lindholm and Frank Yellin, ISBN 0-201-63452-X)에 기술되어 있다.
- <46> 도 8은 도 6에 도시된 브라우저(253)의 상세 구성을 설명하기 위한 블록도이다. 도 8을 참조하면, 브라우저(253)는 HTML 파서(253a), CSS 파서(253b), DOM트리 형성부(253c), 프리젠테이션방식 결정부(253d), 스크립트 인터프리터(253e), UI 제어부(253f) 및 레이아웃 포맷터(253g)로 구성된다.
- <47> HTML 파서(253a)는 HTML형식의 마크업 문서 데이터를 입력받은 경우에 문서가 HTML 문법에 부합하는지 여부를 검증하는 기능을 수행한다.
- <48> CSS 파서(253b)는 마크업 문서 데이터가 색상, 폰트 등에 관한 디스플레이 규칙에 부합하는지 여부를 검증하는 기능을 수행한다.
- <49> DOM(Document Object Model) 트리(Tree) 형성부(253c)는 마크업 문서 데이터를 구조화하는 기능을 수행한다. 즉, 마크업 문서의 헤드부분과 보디부분으로 구분하고, 헤드부분은 다시 타이틀과 스크립트 태그부분으로 구분 등을 하여 마크업 문서 데이터를 트리 형식으로 분류하여 데이터를 구조화한다. 돔 트리 형성부(253c)에서의 구조화 과정에 의해 마크업 문서에 연결된 애플릿이 무엇인지 어디에 저장되었는지 등에 관한 정보가 추출된다.

- <50> 프리젠테이션방식 결정부(253d)는 색상, 폰트 등의 디스플레이 방식을 결정하는 기능을 수행한다.
- <51> 스크립트 인터프리터(253e)는 돔 트리 형성부(253c)로부터 추출된 스크립터(scriptor)를 입력받아 스크립터를 해석하여 소정 명령을 수행한다. 스크립터는 스크립트 언어로 작성된 일종의 명령 프로그램이다.
- <52> UI 제어부(253f)는 사용자와의 인터페이싱을 제어한다.
- <53> 레이아웃 포맷터(253g)는 마크업 문서의 각 태그에 표시된 레이아웃정보에 따라 화면에서의 레이아웃포맷을 결정하여 블렌더(270)로 출력한다.
- <54> 본 발명에서는 종래의 애플릿의 기능을 확장하여 새로운 애플릿을 제안한다. 애플릿을 바운드 애플릿(bound applet)과 언바운드 애플릿(unbound applet)으로 종류를 나눈다.
- <55> 바운드 애플릿(bound applet)은 종래와 동일한 애플릿이다. 즉, 마크업 문서의 소정 태그에 정의되어 있으며 마크업 문서에 종속적이다.
- <56> 언바운드 애플릿(unbound applet)은 최초 실행시에는 소정의 마크업 문서에 정의된 바에 따라 실행되지만 실행된 후에는 그 마크업 문서와는 독립적으로 동작한다. 따라서 언바운드 애플릿의 실행이 정의된 마크업 문서가 언로드(unload)되더라도 이와 무관하게 계속 JVM(251)에서 실행된다. 언바운드 애플릿의 라이프 사이클은 애플리케이션 매니저(255)에 의해 제어된다.
- <57> 도 9는 JVM(251), 브라우저(253) 및 애플리케이션 매니저(255)간의 상호 동작을 나타내는 도면이다.

- <58> XHTML 애플릿은 일종의 언바운드 애플릿이다. 그러나 XHTML 애플릿은 다른 언바운드 애플릿과는 다른 특성을 가진다. 즉, 다른 언바운드 애플릿은 콘텐츠 저장매체(100)나 원격지의 콘텐츠 저장매체(도시되지 않음)로부터 콘텐츠 재생장치(200)로 가져오는데 비해 XHTML 애플릿은 콘텐츠 재생장치(200)내의 메모리로부터 가져온다. 즉 XHTML 애플릿은 콘텐츠 재생장치(200)내의 메모리에 미리 저장되어 있던 애플릿이다.
- <59> XHTML 애플릿은 JVM(251) 초기화 단계에서 실행된다. XHTML 애플릿은 다른 바운드 애플릿 또는 언바운드 애플릿이 마크업 문서의 돔 트리 구조에 액세스하기 위한 인터페이스 기능을 수행한다.
- <60> 브라우저(253)는 바운드 애플릿들의 이벤트를 애플리케이션 메니저(255)로 통지하고, JVM(251)은 언바운드 애플릿들의 이벤트를 애플리케이션 메니저(255)로 통지한다. JVM(251)으로부터 언바운드 애플릿들의 이벤트를 통지받은 애플리케이션 메니저(255)는 해당 언바운드 애플릿의 라이프 사이클을 제어한다.
- <61> 따라서 언바운드 애플릿은 마크업 문서와 무관하게 JVM(251)에서 실행되고 언바운드 애플릿 자체에 정의된 명령에 따라 실행을 중단하는 경우에는 이를 JVM(251)이 애플리케이션 메니저(255)로 통지하고, 애플리케이션 메니저(255)는 언바운드 애플릿의 실행을 중지하거나 메모리에서 삭제할 것을 JVM(251)으로 통지한다.
- <62> 도 10은 바운드 애플릿의 라이프 사이클을 나타내는 도면이다.
- <63> 자바 애플릿의 상태는 보통 4, 5개 상태로 정의된다. 본 실시예에서는 Init, Start, Stop, Destroy의 4가지 상태에 대해 설명한다.

<64> 먼저, 콘텐츠 저장매체(100) 또는 인터넷(500)을 통해 입력된 마크업문서는 캐시메모리(230)에 저장된다. 캐시메모리(230)에 저장된 마크업 문서가 프리젠테이션 엔진(250)으로 입력되어 파싱된 후, 브라우저(253)에 의해 돔 트리 구조가 형성된다. 브라우저(253)는 마크업 문서 내의 소정 태그 내에 기재된 애플릿에 관한 정보를 애플리케이션 매니저(255)로 전송하여 애플릿 실행을 요청한다. 애플리케이션 매니저(255)는 브라우저(253)로부터 수신한 애플릿에 관한 정보에 따라 마크업 문서와 연결된 애플릿을 콘텐츠 저장매체(100) 또는 인터넷(500)으로부터 가져와 캐시메모리(230)에 저장한다.

<65> 바운드 애플릿을 실행(launching)하는 방법은 다수가 있을 수 있으나 본 실시예에서는 두 가지를 설명한다.

<66> 첫째, 마크업 문서 내의 오브젝트(object) 태그에 의해 실행하는 방법이다. 다음은 바운드 애플릿을 정의한 오브젝트(object) 태그의 예이다.

<67>

```
<object code="my_applet.class" codebase="dvd://intdisc/"
archive="my_applet.jar" codetype="application/java">
  <param name="first_parm" value="one"/>
  <param name="second_parm" value="two"/>
</object>
```

<68> "my_applet.jar"라는 archive내의 "my_applet.class"라는 바운드 애플릿을 정의하고 있다.

<69> 둘째, ADF(Applet Description File) 파일에 의한 방법이다. 이 방법은 바운드 애플릿이 여러 개의 마크업 문서에 종속적인 경우에 사용한다. ADF 파일에 의해 실행되는 바운드 애플릿을 플러그 인(Plug-in) 애플릿이라고 한다.

<70> 다음은 플러그 인 애플릿을 정의한 ADF 파일의 예이다.

```

<71> <adf code="flash4.class" codebase="dvd:///intdisc/"
      archive="flash4dec.jar" codetype="application/java"
      plugin="yes"
      mimetype="application/x-shockwave-flash" version="4.0">
      <param name="first_parm" value="one"/>
      <param name="second_parm" value="two"/>
      </adf>

```

<72> 후술하겠지만 언바운드 애플릿도 ADF 파일에 의해 정의되어 실행된다. 언바운드 애플릿에 관한 ADF 파일인가 바운드 애플릿에 관한 ADF 파일인가는 "plugin="yes""라는 정보에 의해 구별된다. "plugin="yes""이면 바운드 애플릿임을 나타낸다.

<73> 애플리케이션 매니저(255)는 브라우저(253)로부터 전술한 바와 같은 오브젝트 태그나 ADF 파일을 받아 애플릿이 바운드 애플릿인지 언바운드 애플릿인지를 판단한 후, 애플릿을 실행시킨다.

<74> 전술한 바와 같이 애플리케이션 매니저(255)는 마크업 문서내의 오브젝트 태그 또는 ADF 파일을 해석하여 마크업 문서에 연결된 바운드 애플릿을 콘텐츠 저장매체(100) 또는 인터넷(500)으로부터 가져와 캐시메모리(230)에 로딩한다.

<75> 그 후, 브라우저(253)는 마크업 문서의 돔 트리 구조 형성이 종료되었음을 애플리케이션 매니저(255)로 통지한다. 애플리케이션 매니저(255)는 JVM(251)으로 init()함수를 호출할 것을 통지한다. JVM(251)이 바운드 애플릿 프로그램 내의 init()함수를 호출하면 바운드 애플릿은 Init 상태가 된다.

<76> Init 상태가 된 후, 애플리케이션 매니저(255)는 마크업 문서가 브라우저(253)에 의해 렌더링 중에 JVM(251)으로 통지하여 JVM(251)이 바운드 애플릿 프로그램 내의 start()함수를 호출하면 바운드 애플릿은 스타트(Start) 상태가 된다.

- <77> 그 후 마크업 문서의 언로드 이벤트가 발생하여 브라우저(253)에서 마크업 문서가 언로드되면 애플리케이션 매니저(255)는 JVM(251)으로 통지하여 JVM(251)이 바운드 애플릿 프로그램 내의 stop()함수를 호출하면 바운드 애플릿은 스탑(Stop) 상태가 된다.
- <78> 만약 마크업 문서가 다시 사용자에게 의해 재방문되어 화면에 디스플레이되면 애플리케이션 매니저(255)는 JVM(251)으로 통지하여 JVM(251)이 바운드 애플릿 프로그램 내의 start()함수를 호출하여 바운드 애플릿은 다시 스타트(Start) 상태로 되고 언로드 이벤트 발생하면 다시 스탑(Stop) 상태가 된다.
- <79> 바운드 애플릿이 메모리 관리 정책에 따라 런타임 시스템(251c)에서 삭제되려면, JVM(251)은 삭제시키기 전에 바운드 애플릿 프로그램 내의 destroy()함수를 호출하여 바운드 애플릿이 디스트로이(Destroy) 상태가 되게 한 후, 런타임 시스템에서 삭제한다.
- <80> 도 11은 언바운드 애플릿의 라이프 사이클을 나타내는 도면이다.
- <81> 언바운드 애플릿의 상태도 바운드 애플릿과 같이 Init, Start, Stop, Destroy의 4가지 상태를 가진다.
- <82> 언바운드 애플릿을 실행(launching)하는 방법은 다수가 있을 수 있으나 본 실시예에서는 그 중 ADF(Applet Description File) 파일에 의한 방법을 설명한다.
- <83> 마크업 문서 내의 오브젝트(object) 태그에 언바운드 애플릿을 정의한 ADF 파일이 정의되어 있다. 다음은 ADF 파일이 정의된 오브젝트 태그의 예이다.
- <84>

```
<object data="dvd://intdisc/my_applet.apm"
type="application/apm" style="display:none;">
</object>
```

- <85> 상기 오브젝트 태그를 보면 "my_applet.apm"이 언바운드 애플릿이 정의된 ADF 파일을 나타낸다.
- <86> 다음은 언바운드 애플릿을 정의한 ADF 파일의 예이다.
- <87>

```
<adf code="my_applet.class" codebase="dvd://intdisc/"
archive="my_applet.jar" codetype="application/java">
  <param name="first_parm" value="one"/>
  <param name="second_parm" value="two"/>
</adf>
```
- <88> "plugin="yes""라는 정보가 없으므로 애플리케이션 매니저(255)는 상기 ADF 파일이 "my_applet.class"라는 이름의 언바운드 애플릿을 정의하는 것임을 알 수 있다. ADF 파일에 정의된 바에 따라 애플리케이션 매니저(255)는 "my_applet.class"라는 언바운드 애플릿을 콘텐츠 저장매체(100) 또는 인터넷(500)으로부터 가져와 캐시메모리(230)에 로딩한다.
- <89> 캐시메모리(230)에 로딩된 후, 바운드 애플릿과는 달리 돔 트리 구조의 형성이 종료할 때까지 대기하는 등의 어떠한 동기화 등을 필요로 하지 않고 언바운드 애플릿이 캐시메모리(230)에 로딩되면 곧 애플리케이션 매니저(255)는 JVM(251)으로 init()함수를 호출할 것을 통지한다. JVM(251)이 언바운드 애플릿 프로그램 내의 init()함수를 호출하면 언바운드 애플릿은 Init 상태가 된다.
- <90> Init 상태가 된 즉시, 애플리케이션 매니저(255)는 JVM(251)으로 통지하여 JVM(251)이 언바운드 애플릿 프로그램 내의 start()함수를 호출하면 언바운드 애플릿은 스타트(Start) 상태가 된다.
- <91> 언바운드 애플릿은 바운드 애플릿과는 달리 연결된 마크업 문서의 언로딩 여부에 관계없이 계속 JVM(251)에서 실행된다. 언바운드 애플릿은 자체에 프로그램된 바에 따라 프로그램 실행 중단을 명령하는 이벤트가 발생하면 자체적으로 실행을 중단한다.

- <92> 프로그램 실행 중단을 명령하는 이벤트가 발생하면 JVM(251)이 애플리케이션 매니저(255)로 통지하고, 애플리케이션 매니저(255)는 JVM(251)으로 stop()함수를 호출하도록 다시 통지하면, JVM(251)이 언바운드 애플릿 프로그램 내의 stop()함수를 호출하여 언바운드 애플릿은 스탑(Stop) 상태가 된다.
- <93> 언바운드 애플릿이 메모리 관리 정책에 따라 런타임 시스템(251c)에서 삭제되려면, JVM(251)은 삭제시키기 전에 언바운드 애플릿 프로그램 내의 destroy()함수를 호출하여 언바운드 애플릿을 디스트로이(Destroy) 상태로 만든 후, 언바운드 애플릿을 런타임 시스템(251c)에서 삭제한다.
- <94> 이하에서는 도면을 참조하여 본 발명에 따른 애플릿 실행 방법을 설명한다.
- <95> 도 12는 본 발명에 따른 애플릿 실행 방법을 설명하기 위한 흐름도이다.
- <96> 애플리케이션 매니저(255)는 브라우저(253)로부터 마크업 문서 내의 소정 태그 내에 기재된 애플릿에 관한 정보 및/또는 ADF 파일을 수신하여 애플릿 실행 요청을 접수한다(제710 단계).
- <97> 애플리케이션 매니저(255)는 브라우저(253)로부터 전송한 바와 같은 오브젝트 태그나 ADF 파일을 받아 애플릿이 바운드 애플릿인지 언바운드 애플릿인지를 판단한다(제720 단계).
- <98> 제720 단계에서의 판단결과, 애플릿이 바운드 애플릿이면 애플리케이션 매니저(255)는 마크업 문서내의 오브젝트 태그 또는 ADF 파일에 정의된 바운드 애플릿을 콘텐츠 저장매체(100) 또는 인터넷(500)으로부터 가져와 캐시메모리(230)에 로딩한다(제731 단계).
- <99> 브라우저(253)는 마크업 문서의 돔 트리 구조 형성이 종료되었음을 애플리케이션 매니저(255)로 통지하고, 애플리케이션 매니저(255)는 JVM(251)으로 init()함수를 호출할 것을 통지

한다. JVM(251)이 바운드 애플릿 프로그램 내의 init()함수를 호출하면 바운드 애플릿은 Init 상태가 된다(제732 단계).

<100> Init 상태가 된 후, 애플리케이션 매니저(255)는 마크업 문서가 브라우저(253)에 의해 렌더링 중에 JVM(251)으로 통지하여 JVM(251)이 바운드 애플릿 프로그램 내의 start()함수를 호출하면 바운드 애플릿은 스타트(Start) 상태가 된다(제733 단계).

<101> 그 후 마크업 문서의 언로드 이벤트가 발생하여 브라우저(253)에서 마크업 문서가 언로드되면 애플리케이션 매니저(255)는 JVM(251)으로 통지하여 JVM(251)이 바운드 애플릿 프로그램 내의 stop()함수를 호출하면 바운드 애플릿은 스탑(Stop) 상태가 된다(제734 단계).

<102> 스탑(Stop) 상태중에 애플리케이션 매니저(255)는 마크업 문서가 다시 사용자에게 의해 재 방문되어 다시 화면에 디스플레이되면(제735 단계) 애플리케이션 매니저(255)는 JVM(251)으로 통지하여 JVM(251)이 바운드 애플릿 프로그램 내의 start()함수를 호출하여 바운드 애플릿은 다시 스타트(Start) 상태로 된다.

<103> 그러나 스탑(Stop) 상태중에 바운드 애플릿이 메모리 관리 정책에 따라 런타임 시스템(251c)에서 삭제되려면, JVM(251)은 삭제시키기 전에 바운드 애플릿 프로그램 내의 destroy()함수를 호출하여 바운드 애플릿을 디스트로이(Destroy) 상태로 만든다(제736 단계). 바운드 애플릿을 디스트로이(Destroy) 상태로 만든 후, 바운드 애플릿을 런타임 시스템(251c)에서 삭제한다.

<104> 한편, 제720 단계에서의 판단결과, 애플릿이 언바운드 애플릿이면 애플리케이션 매니저(255)는 ADF 파일에 정의된 언바운드 애플릿을 콘텐츠 저장매체(100) 또는 인터넷(500)으로부터 가져와 캐시메모리(230)에 로딩한다(제741 단계).

- <105> 캐시메모리(230)에 해당 언바운드 애플릿이 로딩되면 애플리케이션 매니저(255)는 JVM(251)으로 init()함수를 호출할 것을 통지한다. JVM(251)이 언바운드 애플릿 프로그램 내의 init()함수를 호출하면 언바운드 애플릿은 Init 상태가 된다(제742 단계).
- <106> Init 상태가 된 후, 즉시 애플리케이션 매니저(255)는 JVM(251)으로 통지하여 JVM(251)이 언바운드 애플릿 프로그램 내의 start()함수를 호출하면 언바운드 애플릿은 스타트(Start) 상태가 된다(제743 단계).
- <107> 언바운드 애플릿의 실행 중단을 명령하는 이벤트가 발생하면 JVM(251)이 애플리케이션 매니저(255)로 통지하고, 애플리케이션 매니저(255)는 JVM(251)으로 stop()함수를 호출하도록 다시 통지하면, JVM(251)이 언바운드 애플릿 프로그램 내의 stop()함수를 호출하여 언바운드 애플릿은 스탑(Stop) 상태가 된다(제744 단계).
- <108> 언바운드 애플릿이 메모리 관리 정책에 따라 런타임 시스템(251c)에서 삭제되려면, JVM(251)은 삭제시키기 전에 언바운드 애플릿 프로그램 내의 destroy()함수를 호출하여 언바운드 애플릿을 디스트로이(Destroy) 상태로 만든다(제746 단계). 언바운드 애플릿을 디스트로이(Destroy) 상태로 만든 후, 언바운드 애플릿을 런타임 시스템(251c)에서 삭제한다.
- <109> 도 13a 내지 도 13e는 본 발명에 따른 바운드 애플릿과 언바운드 애플릿의 예를 나타내는 도면이다.
- <110> 도 13a는 마크업 문서를 나타낸다. 마크업 문서에는 "Movie", "Commentary" 및 "Feature"의 세 버튼(button)이 있다. 사용자가 아이콘을 "Movie" 버튼에 위치시키면 마크업 문서의 하단에 바운드 애플릿의 형태로 구현된 "Play "Return of Mozart""라는 메시지가 뜬다.

- <111> 도 13b는 사용자가 "Movie" 버튼을 클릭한 경우에 언바운드 애플릿으로 구현된 화면이 디스플레이된 것을 나타내는 도면이다.
- <112> 도 13c는 사용자가 다시 도 13a에 도시된 마크업 문서로 되돌아와 "Commentary" 버튼을 클릭한 경우에 언바운드 애플릿으로 구현된 화면이 디스플레이된 것을 나타내는 도면이다.
- <113> 도 13d는 사용자가 다시 도 13a에 도시된 마크업 문서로 되돌아와 "Feature" 버튼을 클릭한 경우에 언바운드 애플릿으로 구현된 화면이 디스플레이된 것을 나타내는 도면이다.
- <114> 도 13b 내지 도 13d에 도시된 언바운드 애플릿들은 마크업 문서와 독립적으로 JVM(251)에서 실행된다. 도 13a에 도시된 마크업 문서는 브라우저(253)에 의해 실행된다.
- <115> 도 13e는 도 13a의 마크업 문서 및 도 13b 내지 도 13d의 언바운드 애플릿들이 동시에 실행되고 있음을 나타내는 도면이다. 즉, 도 13b 내지 도 13d에 도시된 바와 같은 화면을 언바운드 애플릿으로 구현함으로써 이들 다수의 언바운드 애플릿들을 마크업 문서와 무관하게 JVM(251)에서 계속적으로 실행시킬 수 있게되고 따라서 사용자가 원하는 경우에는 즉시 화면으로 출력할 수 있게 된다.
- <116> 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는, ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광데이터 저장 장치 등이 있으며, 또한 캐리어 웨이브(예를 들어, 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식으로 컴퓨터가 읽을 수 있는 코드로 저장되고 실행될 수 있다.

【발명의 효과】

<117> 이상에서 설명한 바와 같이, 본 발명에 따른 애플릿 실행 장치 및 방법은, 애플릿을 마크업 문서에 종속적인 바운드 애플릿과 마크업 문서에 종속적이지 않은 언바운드 애플릿으로 종류를 나누고 각각의 라이프 사이클을 달리 정의하여 애플릿의 기능을 확장시키고, 특히 언바운드 애플릿을 새로이 정의함으로써 마크업 문서의 전환 여부와 무관하게 계속 동작하는 애플릿을 제공하는 효과를 가진다.

【특허청구범위】**【청구항 1】**

(a) 애플리케이션 매니저가 애플릿 실행을 요청받는 단계;

(b) 상기 애플리케이션 매니저가 상기 애플릿이 바운드 애플릿인지 언바운드 애플릿인지 애플릿 종류를 판별하는 단계;

(c) 상기 애플리케이션 매니저가 메모리에 저장된 상기 애플릿을 버추얼 머신에 로딩하는 단계; 및

(d) 상기 (b)단계에서의 판별 결과, 상기 애플릿이 언바운드 애플릿이면, 상기 애플리케이션 매니저가 상기 버추얼 머신에게 통지하여 상기 애플릿이 초기(Init) 상태가 되도록 제어하는 단계;

(e) 상기 애플릿이 초기(Init) 상태가 된 직후, 상기 애플리케이션 매니저가 상기 버추얼 머신에게 통지하여 상기 애플릿이 스타트(Start) 상태가 되도록 제어하는 단계;를 포함하는 것을 특징으로 하는 애플릿 실행 방법.

【청구항 2】

제1 항에 있어서,

(f) 상기 애플릿이 종료하면 상기 버추얼 머신이 상기 애플리케이션 매니저에게 종료를 통지하고, 상기 애플리케이션 매니저는 상기 버추얼 머신에게 통지하여 상기 애플릿이 스탑(Stop) 상태가 되도록 제어하는 단계; 및

(g) 상기 애플리케이션 매니저는 상기 버추얼 머신에게 통지하여 상기 애플릿이 디스트로이(Destroy) 상태가 되도록 제어하는 단계를 더 포함하는 것을 특징으로 하는 애플릿 실행 방법.

【청구항 3】

제1 항에 있어서,

(h) 상기 (b)단계에서 애플릿 종류를 판별한 결과, 상기 애플릿이 바운드 애플릿이면, 상기 애플릿과 연결된 마크업 문서가 문법에 맞는 형태 구조임이 확인되면 상기 애플리케이션 매니저가 상기 버추얼 머신에게 통지하여 상기 애플릿이 초기(Init) 상태가 되도록 제어하는 단계를 더 포함하는 것을 특징으로 하는 애플릿 실행 방법

【청구항 4】

제3 항에 있어서,

(i) 상기 마크업 문서가 브라우저에 의해 렌더링 중에 상기 애플리케이션 매니저가 상기 버추얼 머신에게 통지하여 상기 애플릿이 스타트(Start) 상태가 되도록 제어하는 단계;

(j) 상기 마크업 문서가 상기 브라우저에 의해 언로드되면 상기 애플리케이션 매니저가 상기 버추얼 머신에게 통지하여 상기 애플릿이 스탑(Stop) 상태가 되도록 제어하는 단계;

(k) 상기 마크업 문서가 다시 디스플레이되면 상기 (i)단계 및 상기 (j)단계를 반복하는 단계; 및

(l) 상기 애플리케이션 매니저는 상기 버추얼 머신에게 통지하여 상기 애플릿이 디스트로이(Destroy) 상태가 되도록 제어하는 단계를 더 포함하는 것을 특징으로 하는 애플릿 실행 방법.

【청구항 5】

외부로부터 입력된 마크업 문서를 저장하는 메모리;

애플릿을 실행하는 버추얼 머신;

상기 메모리로부터 상기 마크업 문서를 입력받아, 상기 마크업 문서에 포함된 상기 마크업 문서와 연관된 애플릿에 관한 정보를 출력하는 브라우저; 및

상기 브라우저로부터 상기 애플릿에 관한 정보를 입력받아 상기 애플릿을 외부로부터 가져와 상기 메모리에 저장하도록 제어하고, 상기 브라우저로부터 상기 메모리에 저장된 상기 애플릿의 실행을 요청받아, 상기 메모리에 저장된 상기 애플릿을 상기 버추얼 머신에 로딩하고, 상기 애플릿이 바운드 애플릿인지 언바운드 애플릿인지 상기 애플릿의 종류를 판별하여, 상기 애플릿이 언바운드 애플릿이면, 상기 자바 버추얼 머신에게 통지하여 상기 애플릿이 초기(Init) 상태가 되도록 제어하고, 상기 애플릿이 초기(Init) 상태가 된 직후, 상기 버추얼 머신에게 통지하여 상기 애플릿이 스타트(Start) 상태가 되도록 제어하는 애플리케이션 매니저를 포함하는 것을 특징으로 하는 애플릿 실행 장치.

【청구항 6】

제5 항에 있어서,

상기 애플리케이션 매니저는 상기 버추얼 머신으로부터 상기 애플릿이 종료되었음을 통지받은 경우에, 상기 버추얼 머신에게 통지하여 상기 애플릿이 스탑(Stop) 상태가 되도록 제어하고, 상기 버추얼 머신에게 통지하여 상기 애플릿이 디스트로이(Destroy) 상태가 되도록 제어하는 것을 특징으로 하는 애플릿 실행 장치.

【청구항 7】

제5 항에 있어서,

상기 애플리케이션 매니저는 상기 애플릿의 종류를 판별한 결과, 상기 애플릿이 바운드 애플릿이면, 상기 브라우저로부터 상기 애플릿과 연관된 상기 마크업 문서가 문법에 맞는 형태 구조임을 통지받으면, 상기 버추얼 머신에게 통지하여 상기 애플릿이 초기(Init) 상태가 되도록 제어하는 것을 특징으로 하는 애플릿 실행 장치.

【청구항 8】

제7 항에 있어서,

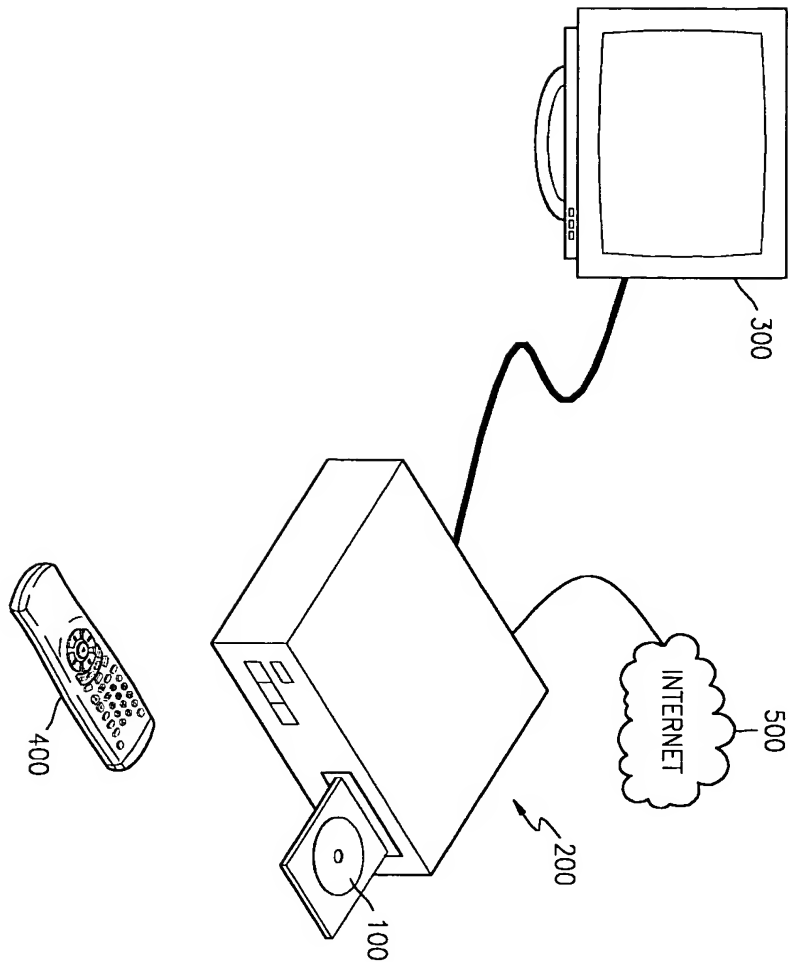
상기 애플리케이션 매니저는 상기 마크업 문서가 상기 브라우저에 의해 렌더링 중에 상기 버추얼 머신에게 통지하여 상기 애플릿이 스타트(Start) 상태가 되도록 제어하고, 상기 마크업 문서가 상기 브라우저에 의해 언로드되면 상기 버추얼 머신에게 통지하여 상기 애플릿이 스탑(Stop) 상태가 되도록 제어하고, 상기 마크업 문서가 다시 디스플레이되면 상기 애플릿이 상기 스타트(Start) 상태 및 상기 스탑(Stop) 상태가 되도록 제어하는 동작을 반복하고, 상기 버추얼 머신에게 통지하여 상기 애플릿이 디스트로이(Destroy) 상태가 되도록 제어하는 것을 특징으로 하는 애플릿 실행 장치.

【청구항 9】

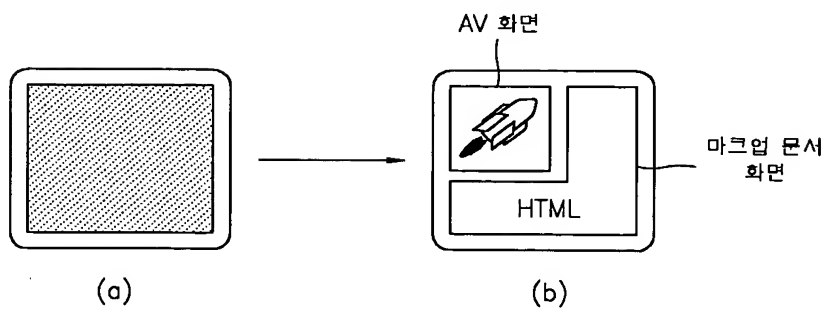
제1 항 내지 제4 항중 어느 한 항에 기재된 방법을 실현하기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

【도면】

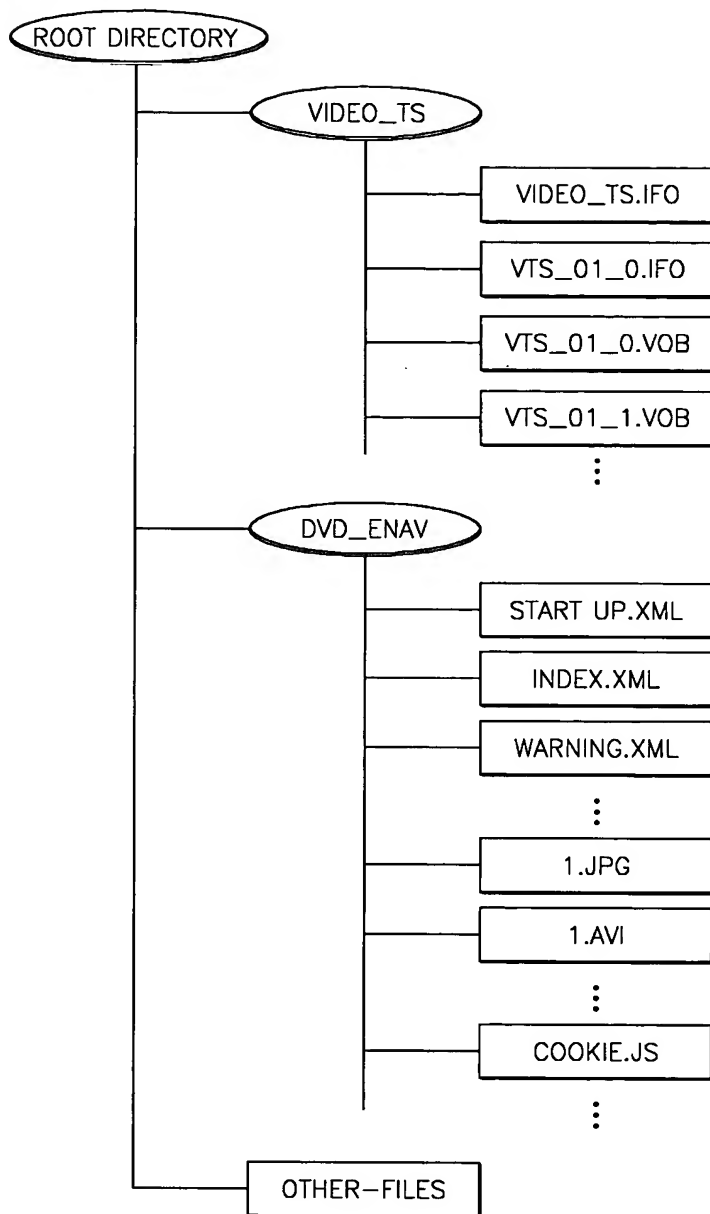
【도 1】



【도 2】



【도 3】

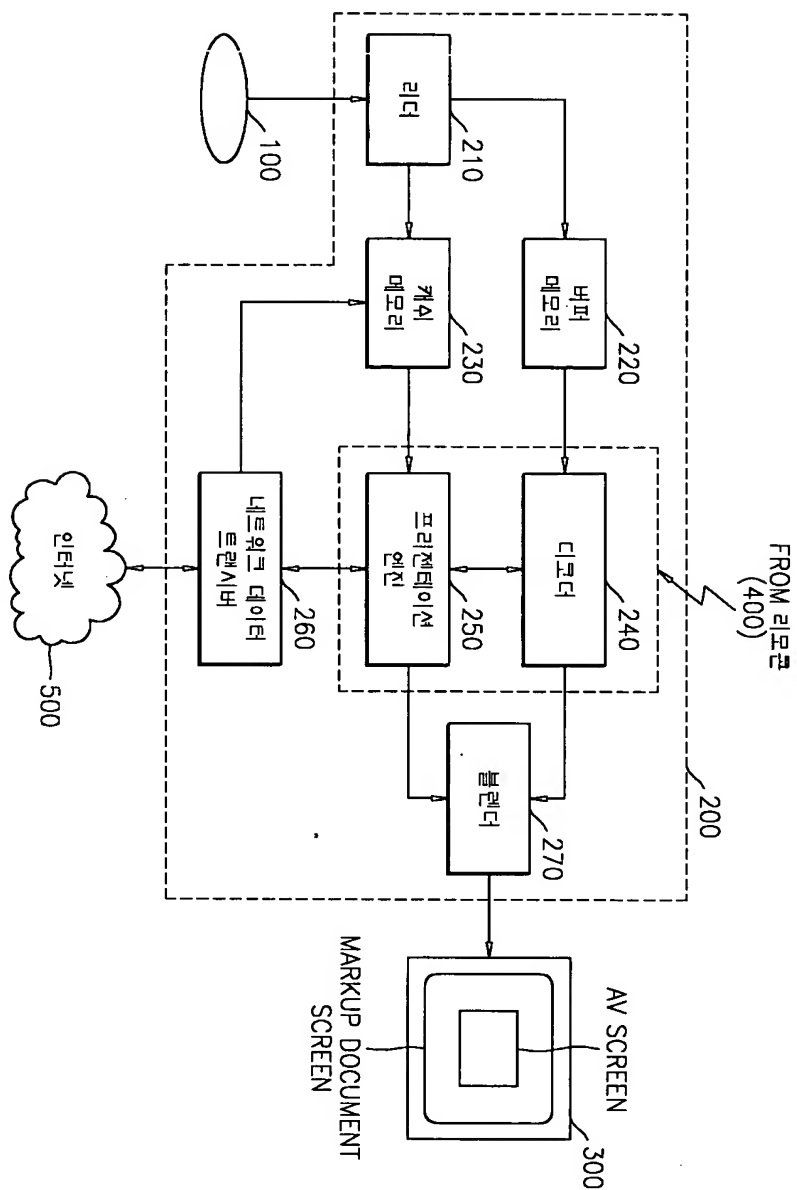


【도 4】

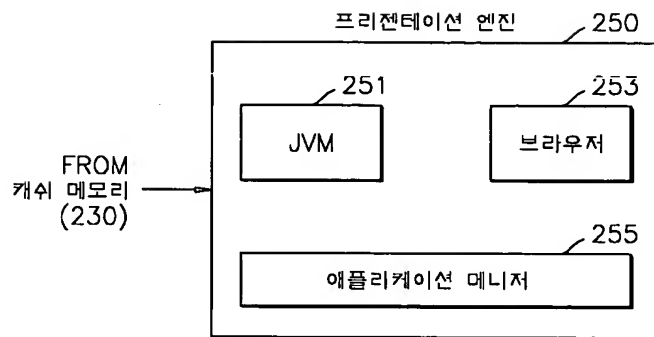
HTML 구성 요소의
예:



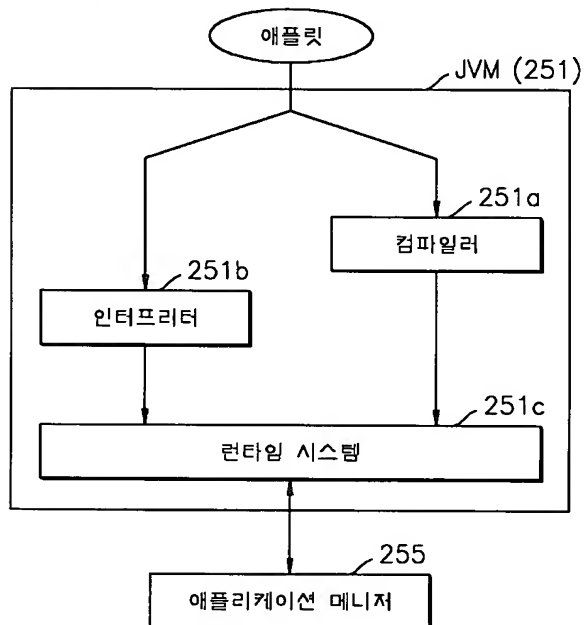
【도 5】



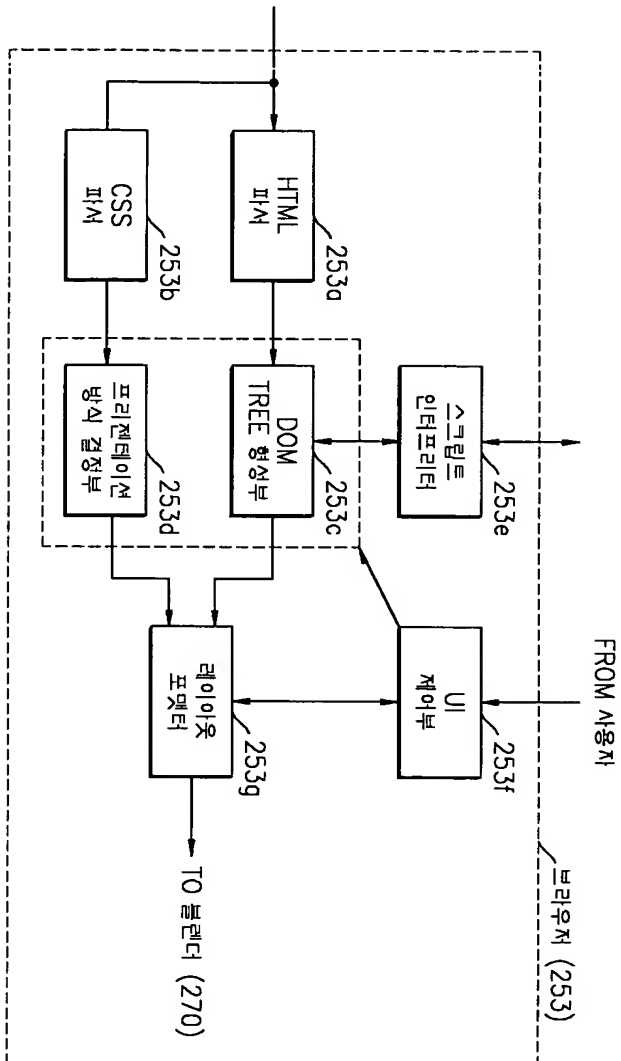
【도 6】



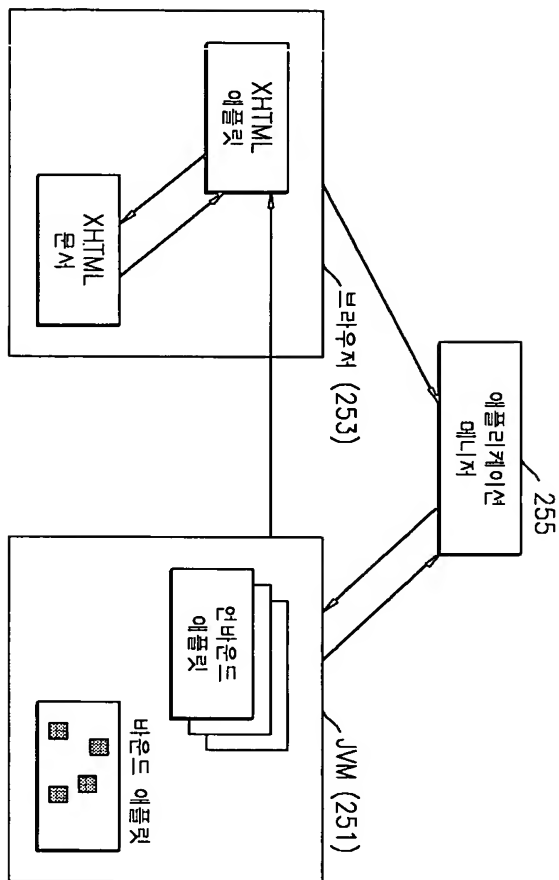
【도 7】



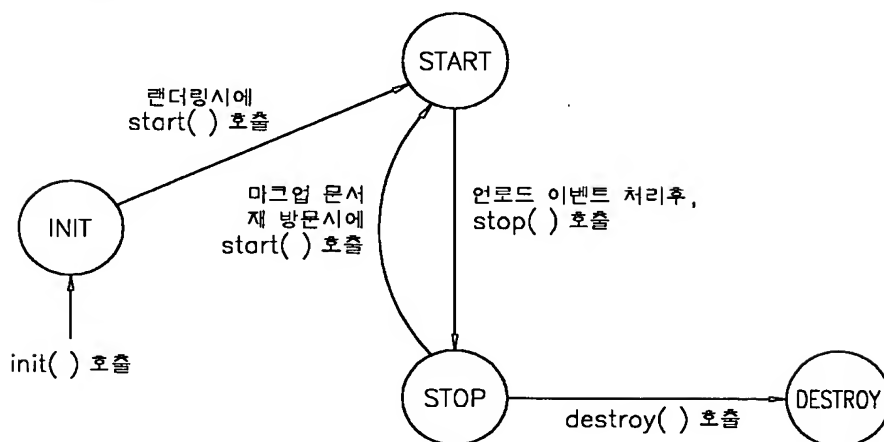
【도 8】



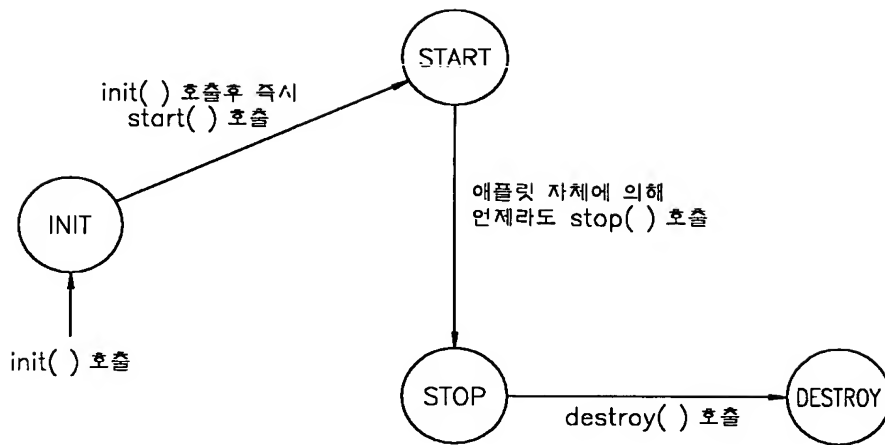
【도 9】



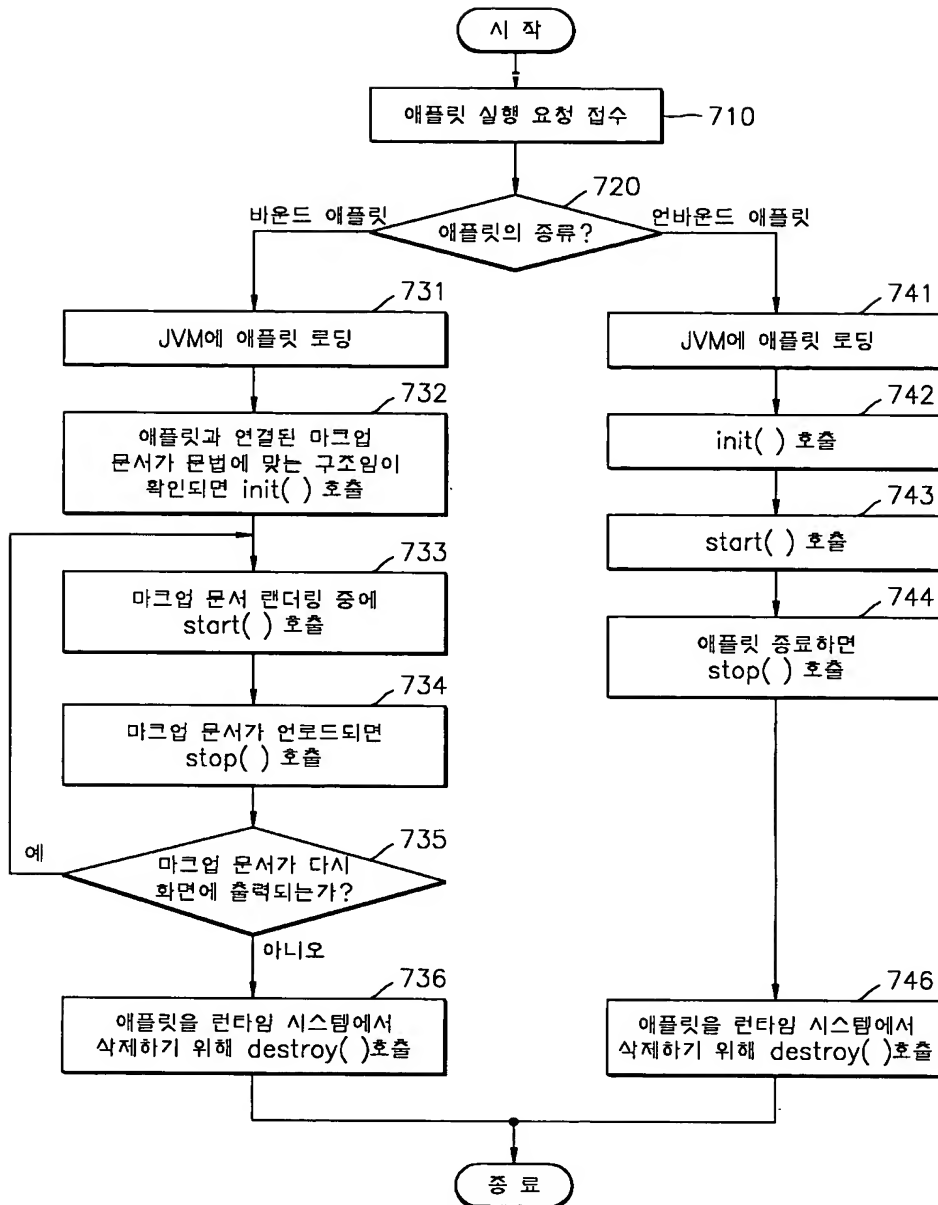
【도 10】



【도 11】

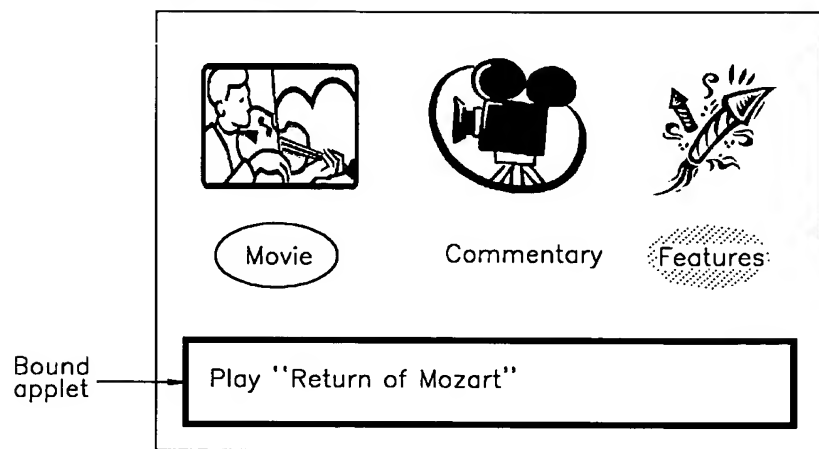


【도 12】



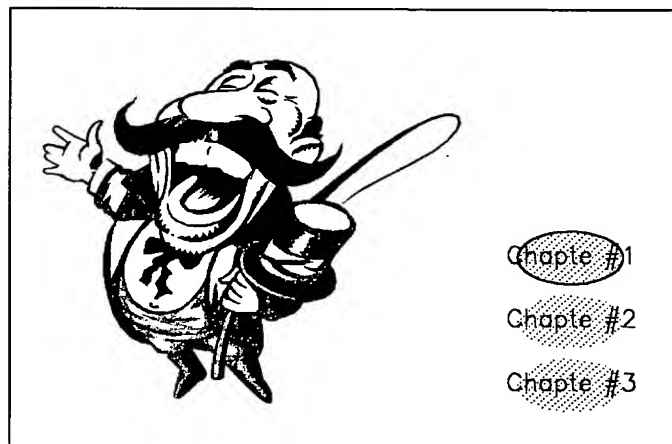
【도 13a】

XHTML page



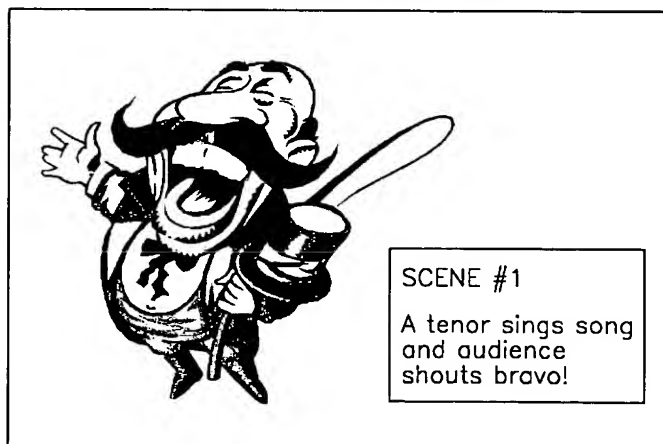
【도 13b】

Unbound Applet for Movie button



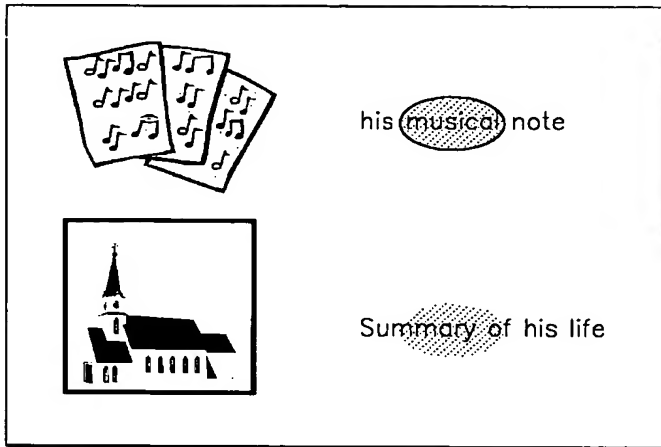
【도 13c】

Unbound Applet for Commentary button



【도 13d】

Unbound Applet for Commentary button



【도 13e】

